修改功能函数：

void modify()

{

// 首先以只读模式打开文件，读取学生信息

    FILE\* file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "r");

    if (file == NULL) {

        perror("无法打开文件");

        return;

    }

    int numStu = 0;

    // 从文件中读取学生信息并存储到 imformation 数组中

    while (fscanf(file, "%d,%[^,],%lf,%lf,%lf,%lf\n",

                &imformation[numStu].num,

                imformation[numStu].name,

                &imformation[numStu].elec,

                &imformation[numStu].expe,

                &imformation[numStu].requ,

                &imformation[numStu].sum) == 6) {

        numStu++;

    }

    fclose(file);

    int modify\_num;

    // 提示用户输入要修改的学生学号

    printf("请输入要修改的学生的学号：");

    scanf("%d", &modify\_num);

    // 标记是否找到要修改的学生

    int found = 0;

    // 以读写模式打开文件

    file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "r+");

    if (file == NULL) {

        perror("无法打开文件进行写入");

        return;

    }

    for (int j = 0; j < numStu; j++) {

        if (imformation[j].num == modify\_num) {

            // 输出当前学生信息

            printf("目前该学生信息：\n");

            printf("学号：%d，姓名：%s，选修课成绩：%.2lf，实验课成绩：%.2lf，必修课成绩：%.2lf，总分：%.2lf\n",

                    imformation[j].num, imformation[j].name, imformation[j].elec, imformation[j].expe, imformation[j].requ, imformation[j].sum);

            // 提示用户输入新的信息

            printf("修改学号为：");

            scanf("%d", &imformation[j].num);

            printf("修改姓名为：");

            scanf("%s", imformation[j].name);

            printf("修改选修课成绩为：");

            scanf("%lf", &imformation[j].elec);

            printf("修改实验课成绩为：");

            scanf("%lf", &imformation[j].expe);

            printf("修改必修课成绩为：");

            scanf("%lf", &imformation[j].requ);

            // 重新计算总分

            imformation[j].sum = imformation[j].elec + imformation[j].expe + imformation[j].requ;

            printf("修改后该学生信息：\n");

            printf("学号：%d，姓名：%s，选修课成绩：%.2lf，实验课成绩：%.2lf，必修课成绩：%.2lf，总分：%.2lf\n",

                    imformation[j].num, imformation[j].name, imformation[j].elec, imformation[j].expe, imformation[j].requ, imformation[j].sum);

            // 将文件指针定位到修改的位置

            //fseek(file, 0, SEEK\_SET);

            for (int i = 0; i < numStu; i++) {

                if (i == j) {

                    fprintf(file, "%d,%s,%.2lf,%.2lf,%.2lf,%.2lf\n",

                            imformation[i].num,

                            imformation[i].name,

                            imformation[i].elec,

                            imformation[i].expe,

                            imformation[i].requ,

                            imformation[i].sum);

                } else {

                    fprintf(file, "%d,%s,%.2lf,%.2lf,%.2lf,%.2lf\n",

                            imformation[i].num,

                            imformation[i].name,

                            imformation[i].elec,

                            imformation[i].expe,

                            imformation[i].requ,

                            imformation[i].sum);

                }

            }

            found = 1;

            break;

        }

    }

    if (!found) {

        printf("未找到要修改的学生学号。\n");

    }

    fclose(file);

}

整体功能概述：

**文件打开与读取**：

首先以只读模式打开文件 D:\\ye\\桌面\\cpp\\school.csv，使用 fopen 函数。

若文件打开失败，通过 perror 输出错误信息并返回。

定义 int numStu = 0 用于统计学生数量，使用 fscanf 从文件中读取学生信息（学号、姓名、选修课成绩、实验课成绩、必修课成绩、总分）存储到 imformation 数组中，每成功读取一条记录，numStu 加 1。

读取结束后关闭文件。

**用户输入：**

提示用户输入要修改的学生学号，并存储在 modify\_num 中。

定义 int found = 0 用于标记是否找到要修改的学生。

**文件修改操作：**

再次以读写模式打开文件 D:\\ye\\桌面\\cpp\\school.csv，若打开失败，输出错误信息并返回。

遍历 imformation 数组，找到与输入学号匹配的学生信息。

若找到，输出当前学生信息，提示用户输入新的信息（学号、姓名、各科成绩），重新计算总分。

输出修改后的学生信息。

将文件指针定位到修改位置，遍历 imformation 数组，将修改后的信息重新写入文件，对于匹配的学生记录，将其新信息写入，对于其他记录，将原信息重新写入。

若找到修改学生，将 found 标记为 1 并退出查找循环。

**结果输出：**

若未找到要修改的学生学号，输出相应信息。

最后关闭文件。

删除学生信息函数：

void del()

{

    // 首先以只读模式打开文件，读取学生信息

    FILE\* file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "r");

    if (file == NULL) {

        perror("无法打开文件");

        return;

    }

    int numStu = 0;

    // 从文件中读取学生信息并存储到 imformation 数组中

    while (fscanf(file, "%d,%[^,],%lf,%lf,%lf,%lf\n",

                &imformation[numStu].num,

                imformation[numStu].name,

                &imformation[numStu].elec,

                &imformation[numStu].expe,

                &imformation[numStu].requ,

                &imformation[numStu].sum) == 6) {

        numStu++;

    }

    fclose(file);

    int del\_num;

    // 提示用户输入要删除的学生学号

    printf("请输入要删除学生的学号：");

    scanf("%d", &del\_num);

    // 标记是否找到要删除的学生

    int found = 0;

    // 以读写模式打开文件

    file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "w");//"w"只写模式。如果文件不存在，创建文件；如果文件存在，清空文件

    if (file == NULL) {

        perror("无法打开文件进行写入");

        return;

    }

    for (int i = 0; i < numStu; i++) {

        // 如果不是要删除的学生，将其信息重新写入文件

        if (imformation[i].num!= del\_num) {

            fprintf(file, "%d,%s,%.2lf,%.2lf,%.2lf,%.2lf\n",

                    imformation[i].num,

                    imformation[i].name,

                    imformation[i].elec,

                    imformation[i].expe,

                    imformation[i].requ,

                    imformation[i].sum);

        } else {

            found = 1;

        }

    }

    fclose(file);

    if (found) {

        printf("学号为 %d 的学生信息已删除。\n", del\_num);

    } else {

        printf("未找到学号为 %d 的学生信息。\n", del\_num);

    }

}

**整体功能概述：**

**文件打开与读取**：

首先以只读模式打开文件 D:\\ye\\桌面\\cpp\\school.csv，使用 fopen 函数。

若文件打开失败，通过 perror 输出错误信息并返回。

定义 int numStu = 0 用于统计学生数量，使用 fscanf 从文件中读取学生信息（学号、姓名、选修课成绩、实验课成绩、必修课成绩、总分）存储到 imformation 数组中，每成功读取一条记录，numStu 加 1。

读取结束后关闭文件。

**用户输入**：

提示用户输入要删除的学生学号，并存储在 del\_num 中。

定义 int found = 0 用于标记是否找到要删除的学生。

**文件删除操作**：

以只写模式打开文件 D:\\ye\\桌面\\cpp\\school.csv，若打开失败，输出错误信息并返回。

遍历 imformation 数组，对于学号不等于 del\_num 的学生信息，将其重新写入文件。

若找到要删除的学生学号，将 found 标记为 1。

**结果输出**：

关闭文件后，根据 found 的值输出相应的信息，若为 1 表示删除成功，输出删除成功信息，若为 0 表示未找到相应学生信息，输出未找到信息。

**交换函数：**

void swap(stu\* a, stu\* b)

    {

        stu temp = \*a;

        \*a = \*b;

        \*b = temp;

    }

学生成绩排序函数：

void order()

{

    // 首先以只读模式打开文件，读取学生信息

    FILE\* file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "r");

    if (file == NULL) {

        perror("无法打开文件");

        return;

    }

    int numStu = 0;

    // 从文件中读取学生信息并存储到 imformation 数组中

    while (fscanf(file, "%d,%[^,],%lf,%lf,%lf,%lf\n",

                &imformation[numStu].num,

                imformation[numStu].name,

                &imformation[numStu].elec,

                &imformation[numStu].expe,

                &imformation[numStu].requ,

                &imformation[numStu].sum) == 6) {

        numStu++;

    }

    fclose(file);

    for (int i = 0; i < numStu; i++)

    {

        for (int  j = 0; j < numStu-1-i; j++)

        {

            if (imformation[j].sum < imformation[j + 1].sum)

            {

                swap(&imformation[j], &imformation[j + 1]);

            }

        }

    }

    for (int i = 0; i < numStu; i++) //打印数组内数据

    {

        printf("学号：%d，姓名：%s，选修课成绩：%.2lf，实验课成绩：%.2lf，必修课成绩：%.2lf，总分：%.2lf\n",

            imformation[i].num, imformation[i].name, imformation[i].elec, imformation[i].expe, imformation[i].requ, imformation[i].sum);

    }

}

**整体功能概述：**

**文件打开与读取**：

首先以只读模式打开文件 D:\\ye\\桌面\\cpp\\school.csv，使用 fopen 函数。

若文件打开失败，通过 perror 输出错误信息并返回。

定义 int numStu = 0 用于统计学生数量，使用 fscanf 从文件中读取学生信息（学号、姓名、选修课成绩、实验课成绩、必修课成绩、总分）存储到 imformation 数组中，每成功读取一条记录，numStu 加 1。

读取结束后关闭文件。

**学生成绩排序**：

调用 swap 函数对学生信息进行排序。

采用冒泡排序算法，通过两层 for 循环，比较相邻学生的总分（imformation[j].sum 和 imformation[j + 1].sum），如果前一个学生的总分小于后一个学生的总分，调用 swap 函数交换它们的位置。

**结果输出**：

最后使用 printf 打印排序后的学生信息，包括学号、姓名、各科目成绩和总分。

**插入学生信息函数：**

void insert()

{

    // 首先以只读模式打开文件，读取学生信息

    FILE\* file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "r");

    if (file == NULL) {

        perror("无法打开文件");

        return;

    }

    int numStu = 0;

    // 从文件中读取学生信息并存储到 imformation 数组中

    while (fscanf(file, "%d,%[^,],%lf,%lf,%lf,%lf\n",

                &imformation[numStu].num,

                imformation[numStu].name,

                &imformation[numStu].elec,

                &imformation[numStu].expe,

                &imformation[numStu].requ,

                &imformation[numStu].sum) == 6) {

        numStu++;

    }

    fclose(file);

    int insert\_num;

    printf("请输入要插入的位置（从 0 开始）：");

    scanf("%d", &insert\_num);

    // 范围检查

    if (insert\_num < 0 || insert\_num > numStu) {

        printf("插入位置无效。\n");

        return;

    }

    // 将插入位置之后的元素向后移动

    for (int i = numStu; i >= insert\_num; i--) {

        imformation[i + 1] = imformation[i];

    }

    // 提示用户输入新的信息

    printf("请输入新的信息：\n");

    printf("学号为：");

    scanf("%d", &imformation[insert\_num].num);

    printf("姓名为：");

    scanf("%s", imformation[insert\_num].name);

    printf("选修课成绩为：");

    scanf("%lf", &imformation[insert\_num].elec);

    printf("实验课成绩为：");

    scanf("%lf", &imformation[insert\_num].expe);

    printf("必修课成绩为：");

    scanf("%lf", &imformation[insert\_num].requ);

    // 计算总分

    imformation[insert\_num].sum = imformation[insert\_num].elec + imformation[insert\_num].expe + imformation[insert\_num].requ;

    // 以读写模式打开文件

    file = fopen("D:\\ye\\桌面\\cpp\\school.csv", "w+");

    if (file == NULL) {

        perror("无法打开文件进行写入");

        return;

    }

    // 将所有学生信息写入文件

    for (int i = 0; i <= numStu; i++) {

        fprintf(file, "%d,%s,%.2lf,%.2lf,%.2lf,%.2lf\n",

                imformation[i].num,

                imformation[i].name,

                imformation[i].elec,

                imformation[i].expe,

                imformation[i].requ,

                imformation[i].sum);

    }

    fclose(file);

    printf("插入成功。\n");

}

**整体功能概述：**

**文件打开与读取**：

首先以只读模式打开文件 D:\\ye\\桌面\\cpp\\school.csv，使用 fopen 函数。

若文件打开失败，通过 perror 输出错误信息并返回。

定义 int numStu = 0 用于统计学生数量，使用 fscanf 从文件中读取学生信息（学号、姓名、选修课成绩、实验课成绩、必修课成绩、总分）存储到 imformation 数组中，每成功读取一条记录，numStu 加 1。

读取结束后关闭文件。

**用户输入**：

提示用户输入要插入信息的位置（从 0 开始），存储在 insert\_num 中。

进行范围检查，若插入位置无效（小于 0 或大于 numStu），输出错误信息并返回。

**元素移动与信息插入**：

将插入位置之后的元素依次向后移动一位，为新元素腾出空间。

提示用户输入新的学生信息（学号、姓名、各科成绩），计算新学生的总分。

**文件更新**：

以读写模式打开文件，若打开失败，输出错误信息并返回。

将更新后的 imformation 数组中的学生信息全部写入文件，使用 fprintf 函数。

关闭文件。

**结果输出**：

输出插入成功信息。